

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES
Docket No. 15065US01**

In the Application of:

Wade Keith Wan, et al.

Serial No.: 10/642,318

Filed: August 15, 2003

For: PSEUDO-RANDOM NUMBER
GENERATION BASED ON
PERIODIC SAMPLING OF ONE
OR MORE LINEAR FEEDBACK
SHIFT REGISTERS

Examiner: Eleni A. Shiferaw

Group Art Unit: 2136

Conf. No.: 2849

Electronically Filed on June 23, 2008

BRIEF ON APPEAL

Mail Stop: Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

The Appellants respectfully request that the Board of Patent Appeals and Interferences reverse the rejections of Claims 1-22 in the present Application. This Brief on Appeal is being filed in response to the Notice of Panel Decision from Pre-Appeal Brief Review dated May 21, 2008.

REAL PARTY IN INTEREST

The real party in interest is Broadcom Corporation, a corporation organized under the laws of the state of California, and having a place of business at 5300 California Avenue, Irvine, CA 92617. Broadcom Corporation is the assignee of the present Application.

RELATED APPEALS AND INTERFERENCES

Not Applicable.

STATUS OF THE CLAIMS

The present Application originally included 24 claims (Claims 1-24). Claims 23-24 were withdrawn. Pending Claims 1-22 stand rejected and are the subject of this appeal. The text of the pending claims and their status is provided in the Claims Appendix.

STATUS OF THE AMENDMENTS

Subsequent to the final rejection mailed on November 26, 2007, no amendments were made.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent Claim 1 is directed to a method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced. The method comprises sampling output sequences of a linear feedback shift register with a specified periodicity. The subject matter of Claim 1 is illustratively described in the present Application at, for example, paragraphs [23], [28], and [29], referring to Figure 3:

[23] Aspects of the present invention may be found in a system and method to generate pseudo-random numbers that are used as encryption keys or seed values in cryptographic applications. The pseudo-random number generator (PRNG) is implemented using one or more linear feedback shift registers (LFSR) that employ a number of techniques to conceal the behavior of its internal parameters or its underlying algorithm. In one embodiment, the outputs of an LFSR are sampled periodically, instead of consecutively at the next iteration, to determine the encryption keys used in the cryptographic application. In one embodiment, the one or more distinct LFSRs are switched periodically after a number of iterations, wherein each of the one or more distinct LFSRs is differentiated by a unique set of feedback parameters or taps. In one embodiment, nonlinear operators are used to map encryption keys generated by a LFSRs to outputs to make it more difficult for a cryptanalyst to decipher the algorithm used in the encryption process. In one embodiment, the configuration of the feedback parameters or taps of a LFSRs are used to determine the initial value of a hashing function used to further encrypt the output generated by the LFSR.

[28] Figure 3 illustrates outputs of a PRNG corresponding to an exemplary $n=3$ bit linear feedback shift register (LFSR), in which the LFSR outputs are sampled every $n=3$ iterations in accordance with an embodiment of the invention. The results of periodic sampling every n iterations is illustrated in Figure 3, in which the LFSR shown in Figure 1 is used and the iterations commence from an initial starting value of (111). As one may see, the use of periodic sampling reduces the correlation between consecutive or

successive outputs of an LFSR.

[29] There are advantages to sampling the LFSR output sequence with periodicity equal to $n=3$. If an n -bit LFSR is sampled once every n iterations then the maximal length properties of its output sequence will be preserved. In addition, sampling once every n iterations prevents revealing the underlying shifting of bits of an LFSR structure, since all n bits related to an encryption key will be processed before the next encryption key is generated. Note that the simplistic implementation of the underlying LFSR is preserved while periodic sampling of the LFSR outputs reduces a cryptanalyst's ability to correlate outputs between consecutive iterations. Although Figure 3 provides an embodiment of a 3 bit LFSR implementation in which outputs are sampled every 3 iterations, it is contemplated that other embodiments may be implemented using a n bit LFSR where $n \neq 3$, in which the output sequence is sampled every n iterations. It is further contemplated that other embodiments may be implemented using an n bit LFSR, in which the output sequence may be sampled with period L , for which $L \neq n$.

The invention of Claim 1 is also described in other parts of the Application, such as in the Abstract and in the Brief Summary of the Invention.

Independent Claim 7 is directed to a method of generating pseudo-random numbers using linear feedback shift registers in which the correlation between successive pseudo-random numbers is reduced. The method comprises periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register. The subject matter of Claim 7 is illustratively described in the present Application at, for example, paragraph [30], referring to Figure 4:

[30] Figure 4 illustrates outputs of a PRNG employing periodic switching between iterative outputs generated by at least a first LFSR and iterative outputs generated by at least a second LFSR in accordance with an embodiment of the invention. The switching is

performed after a specified number of iterations. In embodiments with multiple LFSRs, the switching is performed sequentially from one LFSR to the next as will be described later in Figure 5. Referring to the table shown in Figure 4, the embodiment illustrates switching performed between two exemplary LFSRs. The first LFSR corresponds to the implementation illustrated and previously described in Figure 2. The second LFSR corresponds to a LFSR having feedback taps at bit 0 (LSB) and bit 2 (MSB). For the second LFSR, the feedback taps, again, are modulo-2 summed and fed back to the input of the register corresponding to the MSB, X_2 . In addition to the periodic sampling technique described in the embodiment of Figure 3, the switching technique shown in the embodiment of Figure 4 may foil a hacker's attempts to search the sample space of possible parameter taps of an LFSR. Because, the sample space of parameter taps is often smaller than that corresponding to the sample space of possible encryption keys, a hacker or cryptanalyst may pose a threat if he possesses knowledge of some of the parameters taps or encryption keys. Referring to Figure 4, a simple method to further thwart a cryptanalyst is to continuously switch between LFSRs so that a hacker will be unable to determine an algorithm (that is easily discernible when using a single LFSR). By carefully switching between one or more LFSRs, each configured using M distinct sets of feedback taps, it may be possible to obtain an overall combined output sequence that is periodic with period $M \cdot (2^n - 1)$. The resulting sequence would also have a distribution that is nearly white (or near random) if all the possible values, except the all zero sequence (000), are generated in one period. One configures the M different sets of feedback taps such that each LFSR generates a maximal length sequence, thereby assuring that the output over all LFSRs comprises a nearly white sequence. The only other requirement is that a complete period over the entire set of LFSRs is traversed once and only once every $M \cdot (2^n - 1)$ iterations. There are many ways to accomplish this, but a simple implementation might be to switch from a LFSR (characterized by distinct sets of feedback taps) when a fixed number of iterations is reached; at the same time, store a current state value for the LFSR. The stored state value may be recalled when the algorithm switches back to the LFSR, allowing the LFSR to proceed to the next logical state. The table of Figure 4 shows an example of periodic switching between an exemplary $M=2$ different LFSRs in which the switching is done after every iteration. In fact, the switching may mislead a potential hacker to believe that an algorithm other than LFSRs is being used. Referring to Figure 4, note that the same key may be re-generated after X iterations where $X < 2^n - 1$; however, the

pseudo-random number sequence is not periodic with period $p=X$. This may be seen in Table 3 by noting that the value 100 is generated on iterations 2 and 5; however, the PRNG has period $2*7=14$, and is not periodic with period 3. Again, the use of periodic switching among one or more LFSRs reduces the correlation between consecutive or successive outputs of an LFSR.

The invention of Claim 7 is also described in other parts of the Application, such as in the Abstract and in the Brief Summary of the Invention.

Independent Claim 11 is directed to a method of encrypting a pseudo-random number generated by a linear feedback shift register comprising operating a nonlinear operator on said pseudo-random number and one or more operands. The subject matter of Claim 11 is illustratively described in the present Application at, for example, paragraph [31], referring to Figures 3 and 4:

[31] The behavior of the LFSRs may be further concealed by applying a nonlinear operator such as an exemplary XOR operator to the pseudo-random number generated by the techniques described in Figures 3 and 4. In reference to the LFSR switching technique described in Figure 4, the pseudo-random number may be XORed with a different operand or binary sequence corresponding to each LFSR used. For example, if a 3 bit LFSR is used, a first distinct 3 bit binary number such as (0, 1, 1) may be used as the operand for a first LFSR while a second distinct 3 bit binary number such as (1, 0, 1) may be used as the operand for a second LFSR. If the nonlinear operators are carefully chosen to map every input value to a different output value, then the nearly white distribution of the input to the nonlinear operator will be preserved at its output.

The invention of Claim 11 is also described in other parts of the Application, such as in the Abstract and in the Brief Summary of the Invention.

Independent Claim 17 is directed to a method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function. The method comprises receiving said pseudo-random number generated from

said linear feedback shift register; and varying the initial value of said hashing function over time by way of a function operating on one or more variables. The subject matter of Claim 17 is illustratively described in the present Application at, for example, paragraphs [34-36], referring to Figure 6:

[34] Figure 6 illustrates a functional block diagram of a typical hash (or hashing) function, h , used to further encrypt a pseudo-random number in accordance with an embodiment of the invention. As illustrated, the output of the PRNG can be used as the hash input, x . The hash input, x , is of arbitrary finite length and can be divided into fixed-length r -bit blocks x_i for which $i=1, \dots, t$. This preprocessing, occurring at a preprocessor 604, typically involves appending extra bits (padding) as necessary to attain an overall bit length which is a multiple of the block length r and/or including a block or partial block indicating the bit length of the unpadded input. An internal fixed-size hash function or compression function, f , 608 may be used to compute H_i , a new intermediate result having bit length, n' , for example, as a function of the previous n' bit intermediate result (H_{i-1}) and the input block x_i . The general process of an iterated hash function is shown in Figure 6 in which the input $x = \{x_1, x_2, \dots, x_t\}$. The hashing function may be represented mathematically as follows:

$$H_0 = IV ; H_i = f(H_{i-1}, X_i), 1 \leq i \leq t, h(x) = g(H_t).$$

[35] H_{i-1} serves as the n' -bit chaining variable between stage $i-1$ and stage i , while H_0 is a pre-defined starting value or initializing value (IV). An optional output transformation function, g , 612 may be used as a final step to map the n' -bit chaining variable to an m -bit result $g(H_t)$.

[36] The use of a hashing function for scrambling is well known; however, the initial value of the hashing function (IV) is often a constant value. A simple method to add a time varying element to the hashing function in order to further conceal the hashing function (or underlying algorithm used) is to make the hashing function dependent upon the configuration of the feedback taps of the LFSR used by the PRNG in generating a particular pseudo-random number. For example, the initial value (IV) may be computed as a function, w , operating on a variable such as the configuration of the feedback taps P_I , of its associated LFSR, i.e. $IV = w(P_I)$. The configuration of the feedback taps may vary over

time, for example, when periodic LFSR switching is performed. As a consequence, the initial value (IV) of the hashing function will vary over time and will depend on the LFSR currently used. It is contemplated that the function w may be a function that operates on other variables such as the iteration number or current output state of a LFSR.

The invention of Claim 17 is also described in other parts of the Application, such as in the Abstract and in the Brief Summary of the Invention.

GROUND OF REJECTION TO BE REVIEWED ON APPEAL

I. Claims 1, 3-6, 14-16, and 20-22 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2004/0205095 (hereinafter, Gressel).

II. Claim 1 stands rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 6,993,542 (hereinafter, Meiyappan).

III. Claims 7-10 and 19 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent No. 5,327,522 (hereinafter, Furuta).

IV. Claims 11-13 stand rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2003/0072059 (hereinafter, Thomas).

V. Claim 17 stands rejected under 35 U.S.C. § 102(e) as being anticipated by U.S. Patent Application Publication No. 2005/0066168 (hereinafter, Walmsley).

VI. Claim 18 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Furuta in view of Gressel.

ARGUMENT

In summary, the Appellants respectfully submit that the Board should reverse the rejections to Claims 1-22. Appellants respectfully submit that Claims 1-22 should be allowed because these claims contain patentable subject matter.

I. REJECTION OF CLAIMS 1, 3-6, 14-16, AND 20-22 UNDER 35 U.S.C. § 102(e)

A. Independent Claim 1

Claim 1 is directed to:

1. A method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.

The Examiner has rejected Claim 1 under 35 U.S.C. § 102(e) as being anticipated by Gressel. Without specifically showing how Gressel teaches each and every element, the Examiner simply alleges that Gressel, at the abstract, and at paragraphs [0026-0027], [0046], [0096-0097] teaches “sampling output sequences of said linear feedback shift register with a specified periodicity,” as recited in Claim 1. The Appellants respectfully submit that Gressel, at the abstract, and at paragraphs [0026-0027], [0046], [0096-0097] does not teach “sampling output sequences of said linear feedback shift register *with a specified periodicity*,” as recited in Claim 1 (emphasis denoted in italics).

Gressel, at the abstract, discloses:

A microelectronic apparatus and method for generating random binary words including at least one clocked pseudorandom binary number sequence generator normally operative to generate a cyclic output sequence of binary numbers, each number including a string of binary symbols, the cyclic output sequence including a basic sequence which is generated repeatedly, at least one bit stream generator generating a clocked bit stream including a stream of binary symbols of a first type occasionally interrupted by a binary symbol of a second type, wherein a first varying time interval between the occasional interruptions is intractably correlated to the output sequence of the number sequence generator, wherein each occurrence of an interruption of the stream of binary symbols of the first type by a binary symbol of the second type causes a pseudorandom modification of the cyclic output sequence of the number sequence generator and a sampling device operative to sample the cyclic output sequence of binary numbers thereby to generate a sampled output sequence including at least one sampled binary word.

Based on the foregoing text, nowhere does Gressel, at the abstract disclose anything about “sampling output sequences of a linear feedback shift register with a specified periodicity,” as recited in Claim 1. While the abstract discloses “a sampling device operative to sample the cyclic output sequence of binary numbers” generated by a “pseudorandom binary number sequence generator,” Gressel, at Claim 8, states:

8. A sampling device comprising: an interface for receiving a CPU request to sample an at least pseudorandom binary stream; and a sampler operative to sample the binary stream, responsive to at least one CPU request received by the interface, after a random waiting interval has elapsed.

Thus, based on Gressel, at Claim 8, the sampling device is “responsive to at least one CPU request received by the interface, after a *random* waiting interval has elapsed.” Thus, based on the foregoing evidence, Gressel teaches a sampling device that samples based on receipt of CPU requests. Furthermore, these CPU requests are received after a random time interval. Consequently, the sampling device correspondingly responds after

a random time interval. Thus, nowhere does Gressel teach “sampling output sequences of a linear feedback shift register with a *specified periodicity*,” as recited in Claim 1. In fact, Gressel, teaches away from what is recited in Claim 1, since Gressel’s sampling device samples “after a random waiting interval has elapsed” which is opposite from sampling with a “specified periodicity.”

Therefore, based on the foregoing arguments, Gressel does not teach “sampling output sequences of a linear feedback shift register with a *specified periodicity*,” as recited in Claim 1. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 1. Hence, the Appellants believe that the patentable subject matter recited in Claim 1 should be passed to allowance. Appellants respectfully request the Board to reverse this rejection and allow independent Claim 1. Furthermore, for at least the reason that Claims 2-6, 14-16, and 20-22 depend on independent Claim 1, Claims 2-6, 14-16, and 20-22 should be passed to allowance as well.

B. Dependent Claim 3

Claim 3 is directed to:

3. The method of Claim 1 wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register.

The Examiner has rejected Claim 3 under 35 U.S.C. § 102(e) as being anticipated by Gressel, at paragraph [0175]. Gressel, at paragraph [0175] states:

[0175] Clock: The device, typically an electronic oscillator that generates periodic signals for synchronization of processes. In both preferred embodiments, randomness is typically initiated by simultaneously activating a primary clock, also termed herein a

"system clock", and a second uncorrelated clock, such that randomizing events occur at intractably difficult to estimate intervals. A typical clock cycle occupies a time interval, called a period. Typically, during the majority of the first half of the period the clock cycle signal is stable at a binary one voltage, and during the majority of the second half of the clock period, the voltage is stable at a binary zero level.

The Appellants do not see how Gressel, at paragraph [0175], shows a teaching of "wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register" as recited in Claim 3. Gressel, at paragraph [0175] describes how a "clock" operates and functions to generate periodic signals for synchronization. This has nothing to do with what is recited in Claim 3. Therefore, the Examiner has not shown a teaching of any of the elements recited in Claim 3. For this reason alone, the Appellants request the Board to reverse the Examiner's rejection to Claim 3. Hence, for at least the foregoing reasons, the Appellants submit that Claim 3 is in condition for allowance.

C. Dependent Claims 4-6

Using Claim 4 as exemplary for Claims 4-6, Claim 4 is directed to:

4. The method of Claim 1 further comprising periodically switching between iterative outputs generated by two or more linear feedback shift registers.

The Examiner has rejected Claims 4-6 under 35 U.S.C. § 102(e) as being anticipated by Gressel, at paragraphs [0263-264] and [0281-0282].

Gressel, at paragraphs [0263-264] states:

[0263] The two feedback configurations include: (a) a first configuration with shift register 442 output taps only from flip-flops FF2 and FF5, these output taps also termed herein "feedbacks 2 and 5"; and (b) a second configuration, wherein feedbacks from flip-flops FF3 and FF4 are complemented (added to) feedbacks 2 and 5 by a binary one enabling input on line 410.

[0264] When Random Swap Select on line 410 is a one, AND gate 470 switches in the feedback output from flipflops FF3 and FF4, XORed in exclusive or gate 447, into the results of the output of AND gate, 447. In this four tap feedback configuration, the output from XOR gate 447 is XOR'd by XOR gate 449, to the feedbacks from flip-flops FF2 and FF5. The random swap select on line 410, therefore, transforms the device to a configuration with a single pair feed back to a double pair feedback. The device alternates between one configuration and the other, as the signal on line 410 oscillates.

Based on Gressel, at paragraphs [0263-264], Gressel discloses a single shift register 442 (see Gressel, at Fig. 2). Therefore, Gressel does not teach "two or more linear feedback shift registers," as recited in Claims 4-6. Thus, for at least this reason, the Examiner has not shown a teaching of each of Claims 4-6. Furthermore, Gressel, at paragraphs [0263-264], does not teach "*periodically switching* between iterative outputs generated by two or more linear feedback shift registers," as recited in Claims 4-6 (emphasis denoted in italics). While Gressel describes two feedback configurations of a single shift register and a control line used to alternate between the two feedback configurations of a single shift register, nowhere does Gressel disclose "*periodically switching* between iterative outputs generated by two or more linear feedback shift registers." Hence, for at least the foregoing reasons, Appellants respectfully submit that Claims 4-6 are in condition for allowance.

Gressel, at paragraphs [0281-0282] states:

[0281] The binary contents of each of the nLFSRs 640, 650 and

660 is randomized by two uncorrelated sources. The slip triggers, on lines 622, 624 and 636, emanating from slip trigger generator 670 at staggered instants from slip trigger bus 671, emanate at regular intervals switched in turn in regular intervals, regulated by the fast clock. The average random sequence slip displacement at such triggers is $2^{n/4}$, where n is the number of flip-flops in the nLFSR register. The second source of unpredictability, inherent to each nLFSR, is the change of frequencies of the driving clocks on lines 642, 652 and 662.

[0282] Responsive to each slip trigger command, a corresponding Slip & Mixed Clock Generator 643, 653 or 663 switches the frequency on its corresponding clock line 642, 652 or 662, from the fast clock to the slow clock, for a random interval (a random number of slow clock cycles), as prescribed in the flowchart of FIG. 8A for nLFSR 640. The process described in the flowchart of FIG. 8A for nLFSR 640 may be identical to the random deceleration in nLFSRs 650 and 660. Preferred synchronized timing of the random decelerated clocks generated by clock generators 643, 653 and 663, to avoid glitches, is illustrated in the timing diagram of FIG. 9.

Based on Gressel, at paragraphs [0281-0282], Gressel discloses clocks transmitted via clock lines 642, 652, 662 for clocking linear feedback shift registers (LFSRs) 640, 650, 660, as illustrated in Gressel, at Figure 6. As stated in foregoing paragraph [0282], “corresponding Slip & Mixed Clock Generator 643, 653 or 663 switches the frequency on its corresponding clock line 642, 652 or 662, from the fast clock to the slow clock, for a *random interval* (a random number of slow clock cycles).” (emphasis denoted in italics) Therefore, switching the frequency of clock lines from a fast clock to a slow clock, at a random interval, does not teach “*periodically* switching between iterative *outputs generated by two or more linear feedback shift registers*,” as recited in Claims 4-6 (emphasis denoted in italics). In fact, Gressel, by way of switching the frequency of clock line 642, 652, or 662 “for a *random interval*,” teaches away from “*periodically* switching between iterative *outputs generated by two or more linear feedback shift*

registers,” as recited in Claims 4-6. Hence, for at least the foregoing reasons, Appellants submit that Claims 4–6 are in condition for allowance. Therefore, Appellants request the Board to reverse the Examiner’s rejection to Claims 4-6.

D. Dependent Claims 14-16

Using Claim 14 as exemplary for Claims 14-16, Claim 14 is directed to:

Claim 14 is directed to:

14. The method of Claim 4 further comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

The Examiner has rejected Claims 14-16 under 314 U.S.C. § 102(e) as being anticipated by Gressel, at paragraphs [0217] and [0239].

Gressel, at paragraph [0217] states:

[0217] Nonlinear Feedback Shift Register (nLFSR): Classes of electronic devices wherein the XORed feedbacks from the shift register do not completely determine the sequence of output words. The non-linear methods used in the preferred embodiments, include; a NAND gate to insert a zero into an output sequence when all sensed inputs are one; a NOR gate to insert a one into the next output word, when all sensed inputs are zero; a "slip" pulse which occasionally complements a feedback binary symbol; a control "swap" which alternates the feedback structure thus changing a bit word output sequence.

Thus, Gressel, at paragraph [0217], simply describes the operation of a nonlinear feedback shift register (nLFSR) which is different from “operating a nonlinear operator on said pseudo-random number [generated or output by a linear shift register] and one or more operands,” as recited in Claim 14. Gressel, at paragraph [0217] simply discloses a pseudo-random number being generated from a nonlinear feedback shift register. In other words, Gressel does not teach an operation performed on the pseudo-random

number that is generated from a shift register. By definition, a nonlinear feedback shift register (nLFSR) operates by using nonlinear operators in the feedback loop of the shift register. Thus, Gressel, at paragraph [0217], does not teach “operating a nonlinear operator on said pseudo-random number (generated by a *linear feedback shift register*, since Claims 14-16 depend on Claim 1) and one or more operands,” as recited in Claims 14-16. By way of teaching a “nonlinear feedback shift register (nLFSR), Gressel, at paragraph [0127], teaches away from a “linear feedback shift register,” which is what does the generating of the “pseudo-random number” recited in Claims 14-16. Thus, Appellants believe that Claims 14-16 contain patentable subject matter that should be allowed.

Gressel, at paragraph [0239] states:

[0239] Slip Sequence Function: A function used in both preferred embodiments that causes a pseudo-random jump displacement in a conventional LFSR. The slip is from one the conventional LFSR sequence to another word in the conventional LFSR sequence. XORing a feedback signal with a random pulse of polarity one implements the process. A slip process preferably is enacted at random intervals occurring a plurality of primary clock cycles more than double the length of the generating nLFSR, to typically avert shortened cyclical sequences.

Appellants request the Board to consider Gressel, at paragraph [0188], which describes the term “displacement” as used in paragraph [0239] above:

[0188] Displacement: In the context of "slips" in an LFSR sequence of words, the jump of the normal place in the word sequence caused by the complementing of the least significant (LS) bit of the next word to appear in the sequence. For example, in the sequence 304 of FIGS. 1A-1B, changing the LS bit 1 in index 10 word, 11011, to zero, causes a displacement to index 25 word, 01011.

Therefore, per Gressel, at paragraphs [0239] and [0188], a function that causes a “pseudo-random” jump displacement (i.e., complementing of the least significant (LS) bit of the next word to appear in the sequence,” does not teach “operating a nonlinear operator on said pseudo-random number and one or more operands,” as recited in Claims 14-16. The Appellants respectfully submit that complementing of the LS bit of the next word in a sequence does not teach “operating a nonlinear operator on [a] pseudo-random number.” Further, Gressel does not teach anything about “operating a nonlinear operator on said pseudo-random number and *one or more operands*.” (emphasis denoted in italics)

Therefore, for the foregoing reasons, the Appellants respectfully submit that Gressel does not show a teaching of each and every element of each of Claims 14-16. Thus, the Examiner has not shown a teaching of Claims 14-16. Consequently, the Appellants request the Board to reverse the Examiner’s rejections to Claims 14-16.

II. REJECTION OF CLAIM 1 UNDER 35 U.S.C. § 102(e)

A. Independent Claim 1

Claim 1 is directed to:

1. A method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.

In addition to the rejection of Claim 1 under 35 U.S.C. § 102(e) as being anticipated by Gressel, the Examiner has rejected Claim 1 under 35 U.S.C. § 102(e) as being anticipated by Meiyappan. Per pages 7-8 of the office action dated November 26, 2007, the Examiner states that Meiyappan, at col. 1 lines 19-24 and at the abstract, teaches “in which the correlation between successive pseudo-random numbers is reduced.”

Appellants respectfully disagree that Meiyappan, at col. 1 lines 19-24 and at the abstract, teaches “in which the correlation between successive pseudo-random numbers is reduced.”

Meiyappan, at col. 1 lines 19-24 states:

The cryptographic techniques underlying such secure protocols require generation of random numbers to generate encryption and decryption keys that assure secure operation. The security achieved depends on generation of truly random numbers whose values cannot be predicted by those seeking to compromise security.

Based on the preceding passage, Meiyappan does not disclose anything about reducing the correlation between successive pseudo-random numbers, as recited in Claim 1. While Meiyappan discloses that cryptographic techniques require the generation of random numbers and that the security achieved depends on generation of truly random numbers, Meiyappan does not disclose anything about *reducing the correlation between successive pseudo-random numbers*.

Meiyappan, at the abstract states:

Truly random numbers are generated with a minimum of extra hardware by taking advantage of the inherent noise in a communication channel. Random numbers can thus be generated without specialized manufacturing requirements and can be

incorporated to conventional integrated circuits with minimal additional logic. The random number generation technique offloads the processor from performing extensive generation calculations without the use of the hardware accelerator. This random number generation technique may find application in, e.g., any network device that participates in a virtual private network or is used to implement electronic commerce.

Based on the preceding passage, Meiyappan does not disclose anything about reducing the correlation between successive pseudo-random numbers, as recited in Claim 1. While Meiyappan discloses general information regarding the generation of random numbers, including the benefits and application of random number generation techniques, nowhere does Meiyappan disclose anything about *reducing the correlation between successive pseudo-random numbers*.

Therefore, for at least the foregoing reasons, Meiyappan, at col. 1 lines 19-24 and at the abstract, does not teach each and every element recited in Claim 1. Therefore, the Examiner has not shown a teaching of Claim 1. Therefore, Appellants respectfully request the Board to reverse the rejection to Claim 1 based on this argument.

Per page 8 of the office action dated November 26, 2007, the Examiner further states that Meiyappan, at col. 3 lines 14-32 and at Figure 2 element 206, teaches “said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.” Appellants respectfully disagree that Meiyappan discloses “said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.”

Meiyappan, at col. 3 lines 14-32 states:

Not every N bit sample is used in generating random numbers. At step 206, a sampling switch 110 samples the output of bit reordering block 108. Sampling switch 110 samples during periods

when its sampling input line is active and does not sample when its sampling input is inactive. Sampling switch 110 may be implemented by a simple FET. The sampling input to sampling switch 110 is provided by the output of a linear feedback shift register 112. The internal structure of linear feedback shift register 112 is known in the art. Further details of linear feedback shift register operation are described in Schneier, *Applied Cryptography*, (2nd Ed. 1996), pp. 372-378, the contents of this entire volume being incorporated herein by reference for all purposes.

The effect of reordering block 108 and sampling switch 110 is to remove the non-random structure of the transmitted signal and therefore isolate the noise component. The output of sampling switch 110 is also N bits wide and is periodically clocked into a random number storage register 114 at a step 208.

Meiyappan, at Figure 2, refers to a flow chart in which element 206 comprises one block in the flow chart. Element 206 (or block 206) states "Sample using LFSR output." Based on the preceding passage, Meiyappan does not state or disclose anything about how the output of a linear feedback shift register is sampled. As illustrated in Figure 1 of Meiyappan, the linear feedback shift register is simply input into the sampling switch 110, in which the "sampling switch 110 samples (the N bit sample obtained from the Bit Reorder block 108) during periods when its sampling input line is active and does not sample when its sampling input is inactive." Therefore, Meiyappan does not teach anything about "sampling output sequences of said linear feedback shift register with a specified periodicity," as recited in Claim 1. As stated in Meiyapan, at col. 3 lines 14-32, "[t]he sampling input to sampling switch 110 is provided by the output of a linear feedback shift register 112." Further, Meiyappan states that "the output of sampling switch 110 is also N bits wide and is periodically clocked into a random number storage register 114." Meiyappan merely discloses that the output of a "sampling switch" is

periodically clocked. Therefore this does not teach “sampling output sequences of [a] linear feedback shift register,” as recited in Claim 1. In other words, Meiyappan is different from what is recited in Claim 1 because the N bit samples generated by the bit reorder block 108 are clocked at the output of a sampling switch. This has nothing to do with sampling or clocking *output sequences of a linear feedback shift register* “with a *specified periodicity*.” Thus, based on the preceding argument, Meiyappan does not teach each and every element and/or feature recited in Claim 1. Consequently, Claim 1 contains patentable subject matter which should be passed to allowance.

Based on the foregoing arguments, Meiyappan does not teach “sampling output sequences of a linear feedback shift register with a *specified periodicity*,” as recited in Claim 1. Consequently, the Examiner has not shown a teaching of each and every element recited in Claim 1. Hence, the Appellants believe that the patentable subject matter recited in Claim 1 should be passed to allowance. Appellants respectfully request the Board to reverse this rejection and allow independent Claim 1. Furthermore, for at least the reason that Claims 2-6, 14-16, and 20-22 depend on independent Claim 1, Claims 2-6, 14-16, and 20-22 should be passed to allowance as well.

III. REJECTION OF CLAIMS 7-10 AND 19 UNDER 35 U.S.C. § 102(e)

A. Independent Claim 7

Claim 7 is directed to:

7. A method of generating pseudo-random numbers using linear feedback shift registers in which the correlation between successive pseudo-random numbers is reduced, said method comprising periodically switching between iterative outputs

generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register.

Claim 7 has been rejected under 35 U.S.C. § 102(e), as being anticipated by Furuta. The Examiner alleges that “in which the correlation between successive pseudo-random numbers is reduced, said method comprising periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register,” as recited in Claim 7, is anticipated by Furuta, at col. 67 line 36 – col. 68 line 2.

Furuta, at col. 67 line 36 – col. 68 line 2 states:

FIG. 126 shows an embodiment of the switching circuit 1309. This switching circuit 1309 includes an OR gate 1310, AND gates 1311 and 1312, and an inverter 1313 which are connected as shown. The control signal is input to the terminal 1314, and the bits b₀ and b₆ output from the output parts 1303 and 1304 of the flipflops 1302₁ and 1302₇ are respectively input to terminals 1317 and 1316. An output of the OR gate 1310 is output from a terminal 1318 and is supplied to the input part 1306 of the flip-flop 1302₁ as the output of the switching circuit 1309.

Normally, the switching circuit 1309 selectively outputs the output of the exclusive-OR gate 1305 in response to the control signal. In this case, the connection of the random number generator 331 shown in FIG. 125 is the same as that shown in FIG. 74. But since the random pulses will be repeated periodically if this connection is fixed, this embodiment switches the connection of the switching circuit 1309 in response to the control signal after a predetermined number of bits are shifted in the LFSR 1302.

For example, this predetermined number of bits corresponds to the number of bits which are shifted in the LFSR 1302 during one period of the random pulses. When the connection of the switching circuit 1309 is switched to selectively output the bit b₆ from the flipflop 1302₇, the initial value set in the LFSR 1302 after one period of the random pulses is changed from the original initial value by shifting an arbitrary number of bits in the LFSR 1302. Thereafter, the connection of the switching circuit 1309 is returned

to selectively output the output of the exclusive-OR gate 1305. Therefore, it is possible to guarantee the random nature of the random pulses over a plurality of periods of the random pulses.

As stated in the preceding passage from Furuta, at col. 67 lines 60-65, “when the connection of the switching circuit 1309 is switched to selectively output the bit b6 from the flipflop 1302₇, the initial value set in the LFSR 1302 after one period of the random pulses is changed from the original initial value by shifting an arbitrary number of bits in the LFSR 1302.” Since “an *arbitrary* number of bits” is shifted after connection of the switching circuit 1309 is switched, Furuta does not teach “periodically switching *between* iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register,” as recited in Claim 7. In other words, since “shifting an *arbitrary* number of bits in the LFSR 1302” would take an “arbitrary” number of clock cycles (i.e., an arbitrary amount of time) prior to switching by the switching circuit 1309, Furuta does not teach “periodically switching between iterative outputs” as recited in Claim 17. Therefore, the switching done by Furuta’s LFSR is done at “arbitrary” time intervals. Furthermore, since Furuta discloses only one feedback shift register, Furuta does not teach a “first linear feedback shift register” and a “second linear feedback shift register.” As a consequence, the Appellants respectfully submit that Furuta teaches away from what is recited in Claim 7. Therefore, the Examiner has not shown a teaching of Claim 7. For these reasons, the Appellants request the Board to reverse the Examiner’s rejection to independent Claim 7. Furthermore, for at least the reason that Claims 8-10 and 18-19, depend on independent Claim 7, Claims 8-10 and 18-19 should be passed to allowance as well.

B. Dependent Claims 9-10

Taking Claim 9 as exemplary for Claims 9-10, Claim 9 is directed to:

Claim 9 is directed to:

9. The method of Claim 7 wherein said pseudo-random numbers are generated with period equal to the sum of each of the individual linear feedback shift register periods.

The Examiner has rejected Claim 9 under 35 U.S.C. § 102(e) as being anticipated by Furuta, at col. 47 line 47 – col. 48 line 15. Furuta, at col. 47 line 47 – col. 48 line 15 states:

One neuron unit 50 receives a plurality of input signals, and a plurality of logical products are obtained between the input signal and the weighting coefficient. Hence, the OR circuit 52 obtains a logical sum of the logical products. Since the input signals are synchronized, the logical sum becomes "111000" when the first logical product is "101000" and the second logical product is "010000", for example. FIG. 79 shows the logical products input to the OR circuit 52 and the logical sum $U(y_i \cap T_{ij})$ which is output from the OR circuit 52. This corresponds to the calculation of the sum and the non-linear function (sigmoid function) in the case of the analog calculation.

When the pulse densities are low, the logical sum of such pulse densities is approximately the sum of the pulse densities. As the pulse densities become higher, the output of the OR circuit 52 saturates and no longer approximates the sum of the pulse densities, that is, the non-linear characteristic begins to show. In the case of the logical sum, the pulse density will not become greater than "1" and will not become smaller than "0". In addition, the logical sum displays a monotonous increase and is approximately the same as the sigmoid function.

As described above, there are two types of couplings (or weighting), namely, the excitatory coupling and the inhibitory coupling. When making numerical calculations, the excitatory and inhibitory couplings are described by positive and negative signs

on the weighting coefficient. In this embodiment which uses digital circuits, the couplings are divided into an excitatory group and an inhibitory group depending on the positive and negative signs on the weighting coefficient T it. Then, the calculation up to the part where the logical sum of the logical products of the pulse trains of the input signals and the weighting coefficients are carried out for each group.

As stated in the preceding passage from Furuta, at col. 47 line 47 – col. 48 line 15, “a logical sum of logical products” obtained by using an OR circuit 52 does not teach a “period equal to the sum of each of the individual linear feedback shift register periods,” as recited in Claims 9-10. As illustrated in Furuta, at Figure 20, for example, an OR circuit 52 that is used to logically OR a number of input signals does not teach a “period equal to the sum of each of the individual linear feedback shift register periods.” Thus, the Examiner has not shown a teaching of the elements recited in Claims 9-10. Therefore, for at least these reasons, the Appellants request the Board to reverse the Examiner’s rejection to Claims 9-10.

C. Dependent Claim 19

Claim 19 is directed to:

19. The method of Claim 18 wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register.

Regarding Claim 19, the Examiner states: “Regarding Claim 19, Furuta et al. teaches the method wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register (col. 44 lines 55 – col. 45 lines 32).”

The Examiner has rejected Claim 19 under 35 U.S.C. § 102(e) as being anticipated by Furuta, at col. 44 line 55 – col. 45 line 32, which states:

FIG. 74 shows a first embodiment of the random number generator 331. In FIG. 74, the random number generator 331 includes an exclusive-OR gate 1305 and 7 flip-flops 1302₁ through 1302₇ which are connected as shown. A clock signal is input to a terminal 1307 and is applied to clock terminals CK of each of the flip-flops 1302₁ through 1302₇. The 7 flip-flops 1302₁ through 1302₇ form a 7-bit linear feedback shift register (LFSR) 1302 together with the exclusive-OR gate 1305. An initial value is set in the LFSR 1302, and the LFSR 1302 thereafter repeats a shift operation. As a result, a number from "1" to "127" and excluding "0" is generated once at random within one random number generation period. This random number which is generated from the LFSR 1302 is defined by bits A1 through A7, where A7 denotes the most significant bit (MSB) and A1 denotes the least significant bit (LSB), for example. However, the bits A7 and A1 may respectively denote the LSB and the MSB.

The output of the flip-flop 1032₁ is input to the exclusive-OR gate 1305 in FIG. 74, but the output of any of the flip-flops 1302₁ through 1302₆ may be input to the exclusive-OR gate 1305.

During the backward process, that is, during the learning process, the register 333 stores the weighting coefficient at the time before the learning process is carried out, and the counter 334 is cleared to zero at the time before the learning process is carried out. The error signal pulse sequences which are collected from the neuron unit of the previous stage and are processed in the gate circuit 78 and the frequency dividing circuit 79 shown in FIG. 39, for example, are input as the signals 75 and 76. Hence, a logic operation is carried out based on the signals 75 and 76, the input signal pulse sequence 55, and the random pulse sequence which is output from the comparator 332 based on the weighting coefficient at the time before the learning process is carried out, and a pulse sequence corresponding to a new weighting coefficient is generated as a result of this logic operation. This pulse sequence corresponding to the new weighting coefficient is input to the counter 334 via the gate circuit 80. The counter 334 counts the number of pulses of this pulse sequence, and the counted value of the counter 334 is transferred to the register 333 when the learning process ends. As a

result, the content of the register 333 is updated or corrected.

Based on Furuta, at col. 44 line 55 – col. 45 line 32, the Appellants do not see how this section of Furuta teaches “wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register,” as recited in Claim 19. Appellants respectfully submit that there is no mention of “variables” in this passage. While Furuta, at col. 44 line 55 – col. 45 line 32 discloses a random number generator, a flip-flop, an exclusive-OR gate, a register, a weighting coefficient, a counter, a neuron unit, a gate circuit, a frequency dividing circuit, a logic operation based on two signals, an input signal pulse sequence, a random pulse sequence output from a comparator based on a weighting coefficient, there is no mention of “wherein said one or more variables comprises the configuration of feedback taps,” as recited in Claim 19. Furthermore, the Appellants respectfully submit that Claim 19 is allowable for at least the reason that it depends on an allowable Claim 18, as Claim 18 is allowable for the reasons provided in Section VI, as described later.

For at least the foregoing reasons, the Appellants request the Board to reverse the Examiner’s rejection to Claim 19. Hence, Appellants respectfully submit that Claim 19 is in condition for allowance.

IV. REJECTION OF CLAIMS 11-13 UNDER 35 U.S.C. § 102(e)

A. Independent Claim 11

Claim 11 is directed to:

11. A method of encrypting a pseudo-random number generated by a linear feedback shift register comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

The Examiner alleges that Thomas, at Claim 29, and at paragraphs [0155] and [0213], teaches “comprising operating a nonlinear operator on said pseudo-random number and one or more operands,” as recited in Claim 11. The Appellants respectfully submit that Thomas, at Claim 29, and at paragraphs [0155] and [0213], does not teach “comprising operating a nonlinear operator on said pseudo-random number and one or more operands,” as recited in Claim 11.

Thomas at paragraph [0155], states:

Referring now to FIG. 10, this figure illustrates a submethod 905 for generating non-linear filtered output bits from shift registers. Step 1005 is the first step of the submethod 905 in which a first tap such as tap 735 and a second tap such as tap 740 of the linear feedback shift register 705 in FIG. 7 are selected. Next, a least significant output bit such as 730 is selected. Next, in Step 1015, the output of the first tap 735 and second tap 740 are combined.

Thomas at paragraph [0213], states:

The present invention has an increased encryption key size that reduces the possibility of a successful attack on a communications channel using the encryption key. The present invention also increases the speed at which a key stream is generated. The present invention generates a key stream that is not derived from shift registers possessing linear relationships between feedback taps. The present invention generates a key stream from feedback taps in a non-linear manner which prevents any attacks on the communication channel when the key stream is used to carry information between parties.

Thomas, at Claim 29, states:

29. A system for securing communications channels, comprising:

a register comprising;

a first tap and a second tap for calculating a first value taken between the outputs of the first and second taps, the output between the first tap and second tap comprising a non-linear value;

an output of the register taken between the first value and a third output bit of the register; and

a new bit comprising an operation taken between the taps of the register.

Nowhere is there any mention of any of the elements and/or features of “operating a nonlinear operator on said pseudo-random number and one or more operands.” For example, paragraph [0155] discloses “a submethod 905 for generating non-linear filtered output bits from shift registers.” A method for generating non-linear filtered output bits does not teach “operating a nonlinear operator on [a] pseudorandom number and one or more operands,” as recited in Claim 11. Generating a nonlinear filtered output bits from shift registers does not teach “operating a nonlinear operator on a pseudo-random number (generated by a linear feedback shift register) and one or more operands. Furthermore, Thomas, at paragraph 0213, discloses generating “a key stream from feedback taps in a non-linear matter” which is different from “operating a nonlinear operator on said [a] pseudorandom number and one or more operands.” Generating a stream by way of a nonlinear shift register does not teach “operating a nonlinear operator on a pseudo-random number (generated by a linear feedback shift register) and one or more operands.” Furthermore, Thomas, at Claim 29, discloses an output between a first tap and a second tap (of a register) comprising a non-linear value. The non-linear value is an intermediary output of a register, which is not a “pseudo-random number generated by a linear feedback shift register,” as recited in Claim 11. Furthermore, none of the passages

from Thomas disclose anything about “operating a nonlinear operator on said pseudo-random number *and one or more operands*.” (emphasis denoted in italics) Therefore, for each of these reasons individually, the Appellants respectfully submit that the Examiner has not shown a teaching of what is recited in independent Claim 11. As a consequence, the Appellants believe that Claim 11 contains patentable subject matter. For these reasons, the Appellants request the Board to reverse the Examiner’s rejection to Claim 11. Furthermore, for at least the reason that Claims 12-13 depend on independent Claim 11, Claims 12-13 should be passed to allowance as well.

V. REJECTION OF CLAIM 17 UNDER 35 U.S.C. § 102(e)

A. Independent Claim 17

Claim 17 is directed to:

17. A method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

Claim 17 is rejected under 35 U.S.C. 102(e) as being anticipated by Walmsley 20050066168 A1. Regarding independent Claim 11, the Office Action states:

Regarding claim 17, Walmsley discloses a method of further encrypting a pseudo-random number (par. 0338, 0344, and 0358) generated from a linear feedback shift register (fig. 9) by using a hashing function (0771, and 0774-0775) comprising: receiving said pseudo-random number generated from said linear feedback shift register (0358-0365 and 0942-0934); and varying the initial value of said hashing

function over time by way of a function operating on one or more variables (0358-0365 and 0942-0934).

See Office Action at pages 4-5.

As was previously mentioned by the Appellants in the response dated September 4, 2007, the office action dated June 1, 2007 had made reference to paragraphs 0942-0934 [sic]. Therefore, it appeared that the office action dated June 1, 2007 contained a typographical error. However, the Examiner had not corrected or addressed this issue on the subsequent office action dated November 26, 2007. Consequently, the Appellants have interpreted the Office Action as referencing paragraphs 0942-*0943* (emphasis denoted in italics).

The Examiner alleges that Walmsley, at paragraphs [0358-0365], and at [0942-0943], teaches “varying the initial value of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 17. The Appellants respectfully submit that Walmsley, at paragraphs [0358-0365], and at [0942-0943], does not teach “varying the initial value of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 17.

Walmsley, at paragraphs [0358-0365], states:

[0358] The protocol passes the chosen random number without the intermediate system knowing its value. This is done by encrypting both the random number and its digital signature.

[0359] The protocol has the following advantages:

[0360] The secret keys are not revealed during the authentication process. The time varying random number is encrypted, so that it is not revealed during the authentication process.

[0361] An attacker cannot build a table of values of the input and output of the encryption process. An attacker cannot call Read without a valid random numbers and signature pair encrypted with

the first key. The second key is therefore resistant to a chosen text attack. The random number only advances with a valid call to Test, so the first key is also not susceptible to a chosen text attack.

[0362] The system is easy to design, especially in low cost systems such as ink-jet printers, as no encryption or decryption is required by the system itself.

[0363] There are a number of well-documented and cryptanalyzed symmetric algorithms to chose from for implementation, including patent-free and license-free solutions.

[0364] A wide range of signature functions exists, from message authentication codes to random number sequences to key-based symmetric cryptography.

[0365] Signature functions and symmetric encryption algorithms require fewer gates and are easier to verify than asymmetric algorithms.

Walmsley, at paragraphs [0942-0943], states:

[0942] The Checksum register is a 160-bit number used to verify that K_1 and K_2 have not been altered by an attacker. Checksum is programmed along with K_1 , K_2 and R with the authentication chip's SSI (Set Secret Information) command. Since Checksum must be kept secret, clients cannot directly read Checksum.

[0943] The commands that make use of Checksum are any that make use of K_1 and K_2 -namely RND, RD, and TST. Before calculating any revealed value based on K_1 or K_2 a checksum on K_1 and K_2 is calculated and compared against the stored Checksum value. The checksum calculated is the 160-bit value $S[K_1|K_2]$.

After reviewing Walmsley, at paragraphs [0358-0365] and at [0942-0943], the Appellants do not see how these paragraphs disclose what is recited in Claim 17. Nowhere is there any disclosure of any of the elements and/or features of “varying the *initial value* of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 17 (emphasis denoted in italics). Furthermore, for

example, paragraphs [0358-0365] and [0942-0934] do not disclose anything about an “initial value of said hashing function” used to “further encrypt a pseudo-random number generated from a linear feedback shift register,” as recited in Claim 17. While Walmsley describes a “protocol” that is “able to validate writes and reads of [an] authentication chip's memory space” and a “checksum register” used for verifying checksums against a stored checksum value, nowhere does Walmsley, at paragraphs [0358-0365] and at [0942-0943], disclose “varying the *initial value* of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 17. Thus, for each of these reasons, the Appellants respectfully submit that the Examiner has not shown a teaching of what is recited in independent Claim 17. Therefore, the Appellants request the Board to reverse the Examiner’s rejection to Claim 17.

VI. REJECTION OF CLAIM 18 UNDER 35 U.S.C. § 103(a)

A. Dependent Claim 18

Claim 18 is directed to:

18. The method of Claim 7 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

On page 8, the final office action dated November 26, 2007 states:

“Claim 19 [sic] is rejected under 35 U.S.C. 103(a) as being unpatentable over Furuta et al. (5327522) in view of Gressel et al. 2004/0205095 A1.”

In the first response dated September 4, 2008, the Appellants had indicated that Examiner's reference to Claim 19 was a typographical error since the Examiner's argument actually referred to the elements recited in Claim 18. However, the Examiner has not corrected this issue since the final office action dated November 26, 2007 displays the same error.

Furthermore, regarding Claim 18, page 8 of the final office action dated November 26, 2007 states:

Regarding claim 18, Furuta et al. teaches the method further comprising: receiving said pseudo-random number generated from said linear feedback shift register (col. 44 lines 55-68); Furuta et al. fails to varying the initial value of said hashing function over time by way of a function operating on one or more variables. However Gressel et al. discloses receiving said pseudo-random number generated from said linear feedback shift register (0148, 0156); and varying the initial value of said hashing function over time by way of a function operating on one or more variables (0183, 0197, 0372, and 0455). Therefore it would have been obvious to one having ordinary skill in the art at the time of the invention was made to combine the teachings because they are analogous in LFSR random number generation. One would have been motivated to incorporate the teachings because it would perform verification of initial value.

For at least the reasons that Claim 18 depends on an allowable independent Claim 7, Claim 18 is allowable as well. As a consequence, the Appellants respectfully submit that dependent Claim 18, which depends on independent Claim 7, should be passed to allowance.

The Examiner alleges that Gressel, at paragraphs [0183], [0197], [0372], and [0455], teaches "varying the initial value of said hashing function over time by way of a function operating on one or more variables," as recited in Claim 18. The Appellants respectfully submit that Gressel, at paragraphs [0183], [0197], [0372], and [0455], does

not teach “varying the initial value of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 18.

Gressel, at paragraph [0183], states:

[0183] Coprocessor: In the parallel application of formally hashing the output of a random string, the electronic device that performs the second randomizing process, e.g., a NIST Secured Hash Algorithm-SHA-1 processor.

Gressel, at paragraph [0197], states:

[0197] Hash: A process of converting a larger binary string, typically 10K bits long, divided into blocks 512 or 1024 bits long, processing the result into a much shorter string, typically 128 or 160 bits long. A hash process is typically programmed such that adversaries are unable to replace a valid hashed message with a fraudulent message such that the hashed result might be identical to the valid result. Examples of hash functions are $H=B^2 \bmod N$, wherein B is the input, N is a prime number and the hashed result is H. A state of the art secured hash standard is SHA-1.

Gressel, at paragraph [0372], states:

[0372] An adversary may gather valuable information by probing the fluctuations of power consumption of a microelectronic device performing a confidential process, e.g., the workings of a gaming machine or the electronic signing of a document or a credit card transaction, with a secret key. Typically such adversarial probing may be masked with either random noise or by operating two such confidential processes, concurrently within range of one another. Masking such a confidential process with a noise emulator generating additive current or voltage fluctuations, resembling the normal confidential process noise, deters such adversarial probing. Outputting the 32 bit output of the 2 nLFSRs, as in bus 1725 of FIG. 10 and FIG. 33 directly into a hash module, as in FIG. 33, is a method to add entropy to the output of the random generator. This method concurrently autonomously radiates signal without utilizing computational resources.

Gressel, at paragraph [0455], states:

[0455] FIG. 33 is a simplified block diagram of a preferred embodiment of a random number generating device. The device

includes the device of FIG. 10 and a Secured Hash Standard Coprocessor, operative to receive the output of unprocessed sequences from the two nLFSRs of FIG. 10, operative to compress the data into 160 bit random strings.

The Appellants have reviewed Gressel, at paragraphs [0183], [0197], [0372], and [0455], but do not see how these paragraphs teach “varying the initial value of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 18. While Gressel, at paragraphs [0183] and [0197] defines a “coprocessor”, and a “hash” respectively, nowhere does Gressel disclose anything about “*varying the initial value* of said hashing function over time by way of a function operating on one or more variables.” While Gressel, at paragraph [0372] describes that “an adversary may gather valuable information by probing the fluctuations of power consumption of a microelectronic device performing a confidential process,” nowhere does this passage disclose anything about “varying the initial value of said hashing function over time by way of a function operating on one or more variables.” Further, while Gressel, at paragraph [0455] describes “a simplified block diagram of a preferred embodiment of a random number generating device” which uses “a secured hash standard coprocessor,” nowhere does this passage disclose anything about “varying the initial value of said hashing function over time by way of a function operating on one or more variables.” Therefore, for at least the foregoing reasons, the Examiner has not shown a teaching of “varying the initial value of said hashing function over time by way of a function operating on one or more variables,” as recited in Claim 18. Therefore, the Examiner has not established a prima facie case of obviousness. Consequently, Claim 18

Application No. 10/642,318
Brief On Appeal Dated: June 23, 2008

should be advanced to allowance. The Appellants request the Board to reverse the Examiner's rejection to Claim 18.

CLAIMS APPENDIX

The following claims are involved in this appeal:

1. A method of generating pseudo-random numbers using a linear feedback shift register in which the correlation between successive pseudo-random numbers is reduced, said method comprising sampling output sequences of said linear feedback shift register with a specified periodicity.
2. The method of Claim 1 wherein said linear feedback shift register generates said output sequences corresponding to maximal length sequences.
3. The method of Claim 1 wherein said specified periodicity is equal to the number of bits output by said linear feedback shift register.
4. The method of Claim 1 further comprising periodically switching between iterative outputs generated by two or more linear feedback shift registers.
5. The method of Claim 3 further comprising periodically switching between iterative outputs generated by two or more linear feedback shift registers.
6. The method of Claim 2 further comprising periodically switching between iterative outputs generated by two or more linear feedback shift registers.
7. A method of generating pseudo-random numbers using linear feedback shift registers in which the correlation between successive pseudo-random numbers is reduced, said method comprising periodically switching between iterative outputs generated by at least a first linear feedback shift register and iterative outputs generated by at least a second linear feedback shift register.

8. The method of Claim 7 wherein said linear feedback shift registers comprise linear shift registers capable of generating maximal length sequences.

9. The method of Claim 7 wherein said pseudo-random numbers are generated with period equal to the sum of each of the individual linear feedback shift register periods.

10. The method of Claim 8 wherein said pseudo-random numbers are generated with period equal to the sum of each of the individual linear feedback shift register periods.

11. A method of encrypting a pseudo-random number generated by a linear feedback shift register comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

12. The method of Claim 11 wherein said nonlinear operator comprises an XOR function.

13. The method of Claim 12 wherein said one or more operands comprises one operand comprising a unique bit sequence corresponding to the LFSR currently used to generate said pseudo-random number.

14. The method of Claim 4 further comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

15. The method of Claim 5 further comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

16. The method of Claim 6 further comprising operating a nonlinear operator on said pseudo-random number and one or more operands.

17. A method of further encrypting a pseudo-random number generated from a linear feedback shift register by using a hashing function comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

18. The method of Claim 7 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

19. The method of Claim 18 wherein said one or more variables comprises the configuration of feedback taps associated with said linear feedback shift register.

20. The method of Claim 14 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

21. The method of Claim 15 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

22. The method of Claim 16 further comprising:

receiving said pseudo-random number generated from said linear feedback shift register; and

varying the initial value of said hashing function over time by way of a function operating on one or more variables.

EVIDENCE APPENDIX
(37 C.F.R. § 41.37(c)(1)(ix))

Not applicable.

RELATED PROCEEDINGS APPENDIX
(37 C.F.R. § 41.37(c)(1)(x))

The Appellants are unaware of any related appeals or interferences.

CONCLUSION

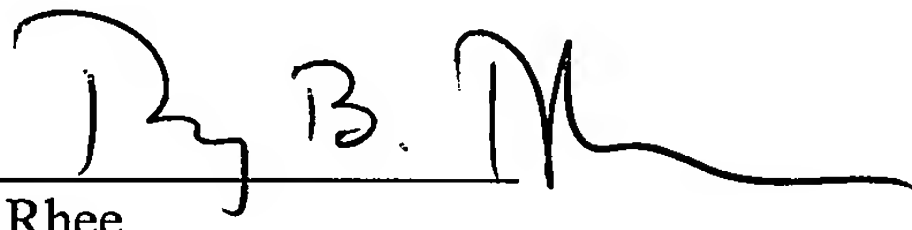
For at least the foregoing reasons, the Appellants submit that Claims 1-22 are allowable in all respects. Reversal of the Examiner's rejections and issuance of a patent on the present Application are therefore requested from the Board.

PAYMENT OF FEES

The Commissioner is hereby authorized to charge \$510 (to cover the Brief on Appeal Fee) and any additional fees or credit any overpayment to the deposit account of McAndrews, Held & Malloy, Account No. 13-0017.

Dated: June 23, 2008

Respectfully submitted,



Roy B. Rhee
Registration No. 57,303

McANDREWS, HELD & MALLOY, LTD.
500 West Madison Street, 34th Floor
Chicago, IL 60661
Telephone: (312) 775-8000
Facsimile: (312) 775-8100